

Learn to Love Support -- they're the best friend you can have in development and planning

Who is this guy

Good morning, everyone! My name is Jerome Comeau, and I've been working in the IT industry (whatever that means) for a little over 20 years. When I was in high school, I read Douglas Coupland's *Microserfs*, a particularly lets-say-rosy view of the life of various people living and working in Tech and specifically in San Francisco, and I decided that I wanted to do **that**, whatever that actually was. I packed up my apartment, sold my car, and I moved 1500 miles to San Francisco basically on spec, knowing no one and having no prospect for a job. I got my first job, as an "overnight monitor" for a startup, when I was 19. The hiring manager for that particular job wanted someone who would stay awake all night and make sure the various tapes got changed and the various blinkenlights stayed green. His point in my interview was, and I quote, "I don't care if you play Doom all night, as long as I don't get paged at 3AM." Since I knew more than ten linux commands, and it was better than my then-current job as a security guard, I said yes,

absolutely, thank you. As the position matured, I learned to handle backups, oracle disk allocation, log rotation, and eventually I was even doing releases and test environment rollouts -- all as a Support person. I've spent time as a sysadmin, a DBA, and a data analyst, but most of my career in Operations has been working with Support teams. I've had experience in small teams, like one and later two people small, and large teams, like 50 or so people, and I've worked with support engineers both on-shore and off-shore, labouring at that interstitial space where people and technology meet, and learning about how Support teams are (but usually aren't) well defined, well-supported, and well-staffed.

What the hell is he talking about

Support is where the rubber meets the road. Failure is inevitable both in people and in software. Planning for that failure requires the input of all the players, so getting the best information as quickly as possible on any potential or actual failure is valuable to both individuals and organizations, and often Support can be the channel through which that failure is noticed and, often, mitigated.

If you're interested in finding trends and use cases and edge cases and pain points in your product, there is no

better voice to have at the table than Support. They can spot the intermittent problems that only crop up every two months but are always fixed by rebooting that one server. They can be extremely clear on the result of the last release, both positive and negative. Often Support is the team most likely to talk to ALL the people involved, including customers, account management, dev and ops. And they can identify the problem in an urgent outage faster than anyone else, because they're able to see and trace the impact of a given failure.

Support is a ready-made pool of people who have intimate knowledge of the product you're building, the organizations you're interacting with, and the customers you're trying to milk for money--er, that is, trying to serve in the most ethical and honest way possible. They are already focused on failure-driven thinking. And they will be extremely grateful for the opportunity to do something that doesn't require a daily grind. For those of you who are not Dev or Ops people, there's a relationship for you and support, as well! Account Management and Sales can get outstanding recruits from Support -- they have intimate knowledge of the platforms, the customers, and (hopefully) the roadmap of features and fixes, and a suite of communication tools that allow them to create and

maintain positive relationships with the users and potential users of the tool.

What is he talking about (pt 2)

So who or what is Support? There's a very broad definition in the Tech industry about what is considered a "support" team and what exactly they're tasked to do, but in general, the Support team is the group of people who are responsible for dealing with the problems of a technological nature that users are having, either by accident or by design. This covers a lot of ground: fixing or provisioning desktop machines for internal employees, responding to account management teams to catalogue their wants and needs, and contacting or accepting contact from external users of a given product to identify and resolve bad user experience, either because the application doesn't work they way they think it should, or because the application isn't working, period. Often Support is tasked with documenting the application and its behaviour, either expected or unexpected, and passing that information on either to users or to the design and development teams via knowledge bases, wikis, or even email. Sometimes Support can be responsible for both internal and external training of users. And frequently,

Support ends up with the various jobs and responsibilities that other teams don't want, but that need to be done and sometimes require access to production systems. For instance, I once worked in a company where the billing for customers was done on a piecework basis: we charged our customers for how much they did using our system. This required backend access, since it was a customer-by-customer report of their system usage for the billing period. Automation was difficult, because inevitably customers had credits or outstanding debits or certain repeating use-cases that weren't included in the billing for whatever reason. Therefore it was the responsibility of Support to generate the report on the specified day and distribute the report to the various account managers to turn into an invoice. The initial report was almost never right because of the adjustments I mentioned before, so frequently we had to run the report over and over again, adjusting as the feedback from the account teams came in and running it again, until the report was considered correct. Even after I started managing I was managing that team, it was such an exceptionally time-intensive job that I never asked anyone else to do it; it didn't seem fair.

In the distant past of the Nineties and the early Oughts, Support was considered an "entry-level" job. It was often a place where someone who had little or no experience with

programming or systems administration could get into the IT industry with a minimal amount of education. Especially in the days when I was getting started in the Tech Startup boom, often the only thing you needed was a recommendation from someone to get an interview. Once you got the job, and if you demonstrated even a little effort or ability, did some time dealing with the various levels of user problems, and particularly were a white male, you got promoted into a department like Systems Administration or Database Management or even Development. From there you could move up the ladder. At one of my previous places of employment, I reported to a person who had dropped out of high school to take a tech job and had worked their way up to Vice President of Operations based entirely on the work experience and connections, and while he was a good boss to me, he was often not a fantastic boss to the team as a whole. Often the idea was that Support was a place for a person to get some job experience and then move on into something bigger and better-paying, which also usually implied that the Support positions weren't particularly well-compensated; the expectation was to work hard and long hours for little pay, and that you could move up if you put in the time; or you'd wash out, and thus weren't a good enough fit for the "culture" of the company, including situations where a person's "culture fit" involved things like skin colour,

reproductive organs, sexual orientation, or particular religious backgrounds.

Support was used as a filter for quote-unquote “good” hires. It was also sometimes used as a pool of cheap labour, and companies could use that expectation of Support compensation to use as downward pressure on the wages of other, more expensive hires like System Administrators and Developers; if you’re living in San Francisco (like I was) trying to get by on 35K per year (like I was), then a promotion that doubles your salary to 70K is a huge opportunity for stability. Nevermind that the Sysadmins who didn’t serve time in Support were hired at 75K per year to start.

The First Middle Bit

The more modern and current view of Support is rather different, but not necessarily better, than the “traditional” view of the job. Today’s Support Engineer role is often recognized as a job-path in and of itself, as the duties usually require a skillset that isn’t particularly valued in other departments, or at least not as valued as other skillsets. To pluck one example from my personal history, when I was in charge of growing my team at a previous place of employment, I went looking for people who were relatively good at so-called “soft” skills: empathy,

communication, responsiveness, ability to listen and learn, and the ability to both read and create documentation. My position at the time was that I could teach anyone the various CLI commands needed to run whatever tools needed to be run, but it was relatively hard to teach a person what was the best way to say “I’m sorry, but I’m afraid I can’t do that” to an upset user who just wanted the tool to WORK, DAMMIT. I don’t think that is the case nowadays, and were I to build a team today, I’d look for people with both strong technical skills and strong customer service experience, because the tools that we support (no matter what they are) are increasingly complicated, technically sophisticated, and often do not fail in particularly intelligent or graceful ways.

In addition to requiring a specific set of skills, because Support has traditionally been a cost-center rather than a revenue generating organization, the drive has been to reduce the cost of support to as close to zero as possible. This led in the mid-oughts to the movement of a significant number of support engineering positions to off-shore locations, most commonly India: a large pool of well-educated English-speaking candidates who were much, much cheaper than a comparable engineer in the United States or Europe. This particular trend has started to reverse itself in the last 5 years or so, as the efforts to

reduce costs led to a raft of anecdotes about language barrier issues and lack of training among the contracted parties. In addition, it's become clear that customer satisfaction and retention are both better for the bottom line, and reducing "churn" in customer contracts can be easily mitigated with a professional and dedicated support organization, especially when coupled with a subscription model of support where contracted support usage can offset some or even all of the cost of the team.

I credit my partner Jean for this thought: no one wakes up in the morning and says "today, I'm gonna contact the Support team, because I really want to have a chat".

Frequently the person contacting Support has absolutely no interest in anything but getting their issue fixed as quickly and as painlessly (to them) as possible, with as little interaction as possible. It's also important to understand that, as Cory Doctorow once said, "the default state of technology is 'broken'" -- that is, all tools require management and maintenance and often fail in new and exciting ways. How that failure is managed by both the tool and the organization can often be the difference between resignation and fury for the user, and many, many, many software tools are not designed to fail gracefully, or even designed to fail at all, despite the fact that it is an inevitability.

Metaphor Alert

A digression: the FAA requires that every plane that flies meet a strict policy on maintenance, from the ultralite built in the garage to 787 Dreamliners built by Boeing. The industry standard is the "five nines", which means that 99.999% of the parts and functionality of the aircraft must be working for the aircraft to be certified as airworthy. If you accept the idea that the average 737 has a million moving parts (in point of fact, there are 1,363,718 moving parts on the average 737-800, including the toilet seats), then that means that every Southwest flight you take there's as many as 10 things on the plane that are broken. The good news is that they often aren't major things -- a seat belt doesn't lock, a cabin compartment doesn't latch, etc. -- but again, the miracle isn't that planes fly, but rather that planes don't fall out of the sky on a regular basis.

Now take into account that the average piece of modern software is much, much more than a million lines of code, and the industry standard for reliability and maintenance are way, way lower than five nines. So the current state of technology, especially enterprise-levelaaS code, often waffles between "on fire" and "no one has noticed we restarted it". And it's Support's job to take the calls and sometimes to make the calls for any and all of those

states. At one of my previous jobs, during incidents that required coordination between departments (like getting the Sysadmin and the on-call developer to talk to each other, for instance), the Support team was responsible for standing up the conference call and bringing all of the parties together online. We were also responsible for keeping notes of the calls, contacting or updating customers impacted by the incident, and then writing up the Root Cause Analysis afterwards. Please note that I said “responsibility”; at this particular job, while Support was responsible for managing the incidents, our team had no actual power to redirect teams or, for that matter, to muster individuals or indicate when there weren’t any team members available to work the problem.

Support is also often the final resting place for all the jobs no one else wants. In one previous job, it was Support’s responsibility to set up, modify, and keep track of changes to client configurations in production. Because the process was so involved, and because people are fallible, there was probably something wrong with one out of every 5 configuration change requests. Eventually the process was semi-automated; it was still done by humans, but it became a checklist selection rather than a process of running copy-pasted curl commands one after the other. At another job it was the Support team’s responsibility to

manually run ad-hoc reports for the Sales and Account teams -- they would provide a list of metrics they wished to see, and it was our job to run a live query in the production database to export the information into CSVs, which we then loaded into excel spreadsheets and created the pivot tables as requested.

The Second Middle Bit

As one previous CEO said (actually, out loud, in a company meeting): when dealing with support and maintenance, “labour costs are fixed, and man-hours are infinite.” So let’s just posit for a moment that your CEO is slightly better than the person I mentioned, and doesn’t believe that people are interchangeable cogs to be used until they’re worn out and then discarded and replaced. That doesn’t necessarily mean that the relationships that other departments have to Support are in any way healthy.

The relationship of Support to the QA team, the Development groups, the Operations groups, and even the Service and Sales groups are traditionally pretty hostile. Both Dev and Ops see support as the jerks who wake them up in the middle of the night, sometimes for good reasons, but sometimes for bad reasons, and who are always asking silly questions like “how does [insert programming or system administration tool here] work”. If

there is a QA team, then often they're the ones who have already pointed out the problems that Support is complaining about. The Customer-facing groups, variously described as Account Management, Account Services, or the like, often don't like Support in the abstract because when they ask Support for help or report a problem, it takes forever to get an answer, and if the customer calls Support directly then inevitably it gets kicked over to Service and then Services has to get on a call with the customer and that's a big interruption to an already busy day. And of course Sales hates Support because it's no big deal to make a promise to, say, use the client's ticketing system instead of the current internal tool; it's a big contract with a long tail, and they shouldn't be moaning about it, what's the big deal.

Support, on the other hand, sees Development as the dudes who always break everything and then don't answer the phone, Operations are the folk who think they're above dealing with Support's complaints, QA as an appendage of Dev and not a "real team", Account Services are the group who doesn't understand how the tool works so they always tell the users to file a ticket for everything, and Sales keeps promising the impossible and never has to deal with the fallout.

OK, these are gross, or rather “wilde” exaggerations, yes. But there’s an undercurrent in the tech industry that’s really rather insidious, which posits that Support people in particular are effectively disposable cogs that can be paid rock bottom prices and are easily replaced. The IT CROWD vision of a bunch of slackers in the basement lording it over the users and playing videogames instead of doing their job is as ubiquitous as the grumbly overweight neckbeard sysadmin with the ponytail and the sandals who growls anytime they’re approached in person and only talks via IRC. That’s not what Support (or systems administration) is in the modern world. The modern Support team member is usually much like the modern Development or Operations folk: they’ve had either experience in the IT field or a degree of some kind, often in Computer Science. Unlike Dev and Ops teams, where it’s not exactly unusual for team members to have spent time in other organizations before moving to their current role, the Support team member is relatively binary: either they’re individuals for whom Support is their first job, or folks that have been in Support for most of their career.

Support is a professional team. These people are not the IT Crowd. They’re also not a phone bank of untrained offshore resources: modern support teams often do have off-shore resources to help to expand support coverage to

24 by 7, especially in the world of As-a-Service software. But those offshore resources are often both more experienced and more educated than on-shore equivalents. In one of my previous positions, the on-shore requirements were pretty standard for support: degree or equivalent experience, plus particular skills. The offshore requirements were much higher: degree required in CS, advanced degree preferred, AND 3-5 years of experience. We also hired new graduates in bulk straight out of college, a dozen at a time, with the understanding that only 4-6 positions would be permanent after 6 months. So if you have a moment, take it easy on the Indian person filing tickets with your engineering team; they're in a cutthroat competition for the chance to maintain employment.

The Final Middle Bit

Most of you probably aren't going to be running your own companies that require Support Teams at scale, but just in case I'll add a rant at the end for those folk. But right now, I want to talk about what you, as developers and as operational professionals, can do to create a good rapport that will help you and help your local support person at the same time.

First: Empathy. Please try to remember that however stressful and overworked you are as a DevOps person, it's even odds that the Support person who's asking you questions is probably more stressed and more overworked. Servers are deployed, restarts are done, releases are either complete or rolled back, but the customer tickets never, ever stop. There is always another phone call from someone who is having another problem and even if no one emailed or called support (which never happens), then there's all of those tickets that are waiting on updates from someone else and need to be communicated in a reasonable manner, hopefully without making the customer too unhappy in the process.

Second: Education. The more you teach your Support cohorts, the less you have to do. Automation is key in every realm, and sometimes that automation can be for another group so it bypasses you completely. If six different support reps ask you about the same issue, that might be an indication that you should take a half-hour and do a knowledge transfer with the support team, so they know what's going on and what needs to be done. Also, if you're a developer, take the time to WRITE DOWN your requirements if someone is assigning you a problem for fixing. If you put it in writing, that allows the support team to refer to an actual template for the information you need.

Third: Escape. Many support reps, even if they love support and want to do it forever, are looking for something else that doesn't result in migraines and ulcers and anxiety medications. They may love parts of the job, but often they are ground down by the unrelenting stream of negative responses and increasing metrics to match; 50% of members of a support organization report having migraines in the first 6 months of work, even if they had no previous experience with migraines. So take the time to find support reps who are clever and interesting and that you like working with, and when the opportunity arises suggest to your manager that this person would be a great opportunity for an associate position that will be able to jump in and get up to speed more quickly than an outside hire.

Fourth: Inclusion (which ruins my pattern but I couldn't think of a good word that starts with E). Support is the group who has their metaphorical finger in every pot. They're creating and responding to Engineering tickets, they're interacting with Operations on a multiple-times-a-day frequency (especially when the product is *aaS). They are taking reports of issues from both internal and external customers. And they would love to share any and all of this with you, either as an informal

knowledge transfer or a more structured scheduled process. So bring them into your stand ups and your Agile groups and even your sprint meetings; they can be a great resource of what is or isn't possible or what is or isn't working.

And for the CEOs and CTOs out there, whether you run big organizations and make long-term decisions not just for yourself but for your teams, or the one-body or two-body operations that are looking to grow into something big, I have this to say:

Metaphor Alert Part 2

Metaphor alert! Think of your business as a bedroom -- let's say you're a homeowner and you're looking to rent out your spare bedroom on AirBnB or something like that. Your bedroom is a business, and your production environment is the bed -- mattress (front-end), box-spring (back-end), bedframe (infrastructure). Your Operations team is the person who changes the sheets (product release), and your Support team are the folk who answer the phone.

If you're running a shady, quasi-illegal operation out of your spare bedroom, the person who changes the sheets

is probably you, and you're probably not a professional housekeeper. You just want clean sheets that keep the mattress from getting horked up by the weirdo from Brooklyn with the Macbook Pro who leaves beard trimmings in the sink. In this case, you do what any reasonable homeowner does: you go out and buy a set of sheets off the shelf, throw on the fitted sheet, and ignore it until the next person comes along and you have to change the sheets again. You're trying to make some spare scratch on the side, not make a business of it, so this model is fine; you can probably get by with two or three sheet sets and you just pull them off and toss them in the laundry as needed, and most of the time you keep your treadmill with the hangers on it in the corner and there's no problem. You answer the phone yourself.

But then you've got some spare cash, so you take out a mortgage on a condo in a building in downtown Portland and rather than moving into it you stage it and decide to rent it out to people visiting PDX for conferences or vacations or whatever, because there's money to be made with spare bedrooms. Now you have a choice: do you become an expert at cleaning? Or do you hire a cleaning service to keep your condo clean between visits? Note that the cleaning service is going to cut into your profits, probably pretty significantly. But you're also going to

spend a lot of time and effort on sheets. And if sheets aren't something you want to spend a lot of time on, especially fiddling with fitted sheets on a given mattress, then there's a pretty steep opportunity cost there as well. So either you get to become an expert on sheets and making the bed, or you're going to spend a moderate chunk of the money you're making to have someone else come in and change your sheets for you. Your choice.

But then you realize, you really like managing visitors, and there's lots and lots of people wanting to sleep in Portland, so that's it: you're going to build a hotel in Portland. You're going to have lots and lots of mattresses for lots and lots of visitors. And that means lots and lots of sheets. So now it's time to make some decisions about hiring the people who know something about sheets, and vacuums, and washing machines, and answering phone calls and taking reservations.

It was pointed out to me by a hospitality specialist that when you're managing a hotel at scale, no one uses fitted sheets. Instead, the proprietor goes to a special wholesaler and buys a metric ton of flat sheets of a uniform colour and size, which the staff then folds and fits to the particular mattress as necessary based on size and

usage. I'll leave the parsing of that as a metaphor for Operations Teams as an exercise for the class.

Like in the hospitality industry, in IT the people who change the sheets and vacuum the floors and fold the corners and spray for bedbugs are fantastically undervalued for the work they do, mostly because when they do it correctly no one notices and when they do it badly companies go under. And often Support is the night auditor: the person working the front desk, taking the phone calls and responding to everything from room service requests to broken plumbing, and frequently getting yelled at by customers about something over which they have absolutely no control.

This metaphor is getting a little out of control, but I hope you get my point: trust people to know what they're doing, let them do it, and for the gods own sake, pay them reasonably well, and give them the chance to grow and expand both their responsibilities and their opportunities, or they will desert you in droves the moment that someone else offers them a dollar more an hour to change the sheets or answer the phones.

I hope you've enjoyed my talk!

As you might have guessed, I'm a big supporter of Operational Thinking, and I've spent most of my career thinking about how to make my organizations better. And as a person who likes the Support role enough to effectively make it a career, I hope I've helped make it understandable to you as QA, Dev, Ops, or DevOps folk why Support is a valuable voice to have at the table and a good friend to have at your back.

Make friends with Support. We won't bite, honest!